Optimal peptide quantification via machine learning enhanced fragment ion ranking in DIA-MS proteomics

BINP51, Bioinformatics: Master´s Degree Project, 45 credits

Lina Lu[1*]

Supervisor: Lars Malmström[1]
Co-supervisor: Aaron Scott[1]

1 Department of Clinical Sciences, Lund University, Klinikgatan, BMC D13, SE-22184, Lund, Sweden

* li7186lu-s@student.lu.se

Abstract

In a standard mass spectrometry workflow, acquired mass spectra are searched against a library of peptides to extract peptide spectra matches (PSMs). Peptides are normally quantified by aggregating the intensities of fragment ions extracted from MS/MS spectra. However, quantifying peptides by summing up the intensities of all fragment ions in PSMs can resulting in inaccurate results due to the fact that multiple fragment ions can interfere with each other in complex samples. This project aims to use machine learning to enhance fragment ion ranking for every precursor to ensure optimal peptide quantification. Here, we describe a workflow that leverages machine learning to pick only the most confident fragments extracted for each potential peptide for quantification. We demonstrate the usability of the workflow on yeast standard benchmark data, showing that the average accuracy of quantification and differential expression across all optimized methods is 22.46% higher than the standard workflow. In addition, we investigate the performance of our workflow on existing complex and low-fold-change proteomic data containing four species and demonstrate the generalizability of our models on unrelated diverse data sets.

Introduction

Mass spectrometry proteomics

Liquid chromatogram-mass spectrometry(*LC-MS*) has changed the way proteomics data is acquired. It is known as a hypothesis-free discovery method that allows the acquisition of data on proteins, peptides and fragment ions without relying on antibodies [1]. In a standard *LC-MS* proteomics workflow of any experiments, proteins are first extracted from the sample using biochemical methods, then cleaved into peptides. Peptides are then passed through the liquid phase and are separated by chromatography and ionized by electrospray ionization into precursor ions. Precursor ions are sent to a mass spectrometer to measure and report the mass-to-charge ratio (m/z) of both the peptide precursor ions and the fragment ions produced by collision of precursor ions [2].

LC-MS greatly improves the efficiency of proteomic data acquisition, making it possible to identify and quantify all proteins in a single run from samples containing thousands of proteins. Several different mass spectrometry-based proteomics strategies have been developed over the past few years [3]. There are currently two main methods of mass spectrometry-based protein research, that are discovery-based qualitative protein analysis and targeted protein quantitative analysis.

In discovery-based qualitative protein analysis, the mass spectrometer automatically selects peptide ions for fragmentation based on signal strength. Match the experimental spectrum to the library spectrum to infer the peptide sequence and thus identify the protein. Whereas in targeted protein quantitative analysis, the mass spectrometer is programmed to detect specific peptide ions from the target protein, with the aim of identifying the protein and accurately quantifying a selected few proteins of interest with high sensitivity and reproducibility.

Data-dependent acquisition (DDA) is also known as shotgun proteomics. It is a discovery-based proteomics mode where the goal is usually to identify as many proteins as possible. In this method, proteins are digested into peptides by typsin or other enzymes. The mass spectrometer then sequentially examines all peptide ions eluting from the liquid chromatography(LC) column at a specific time, conducts a MS1 scan and picks out those precursor ions with the strongest intensity. Those precursor ions are separated and fragmented to generate the MS/MS (MS2) spectrum. DDA is the most used mode for many years, but it only selects peptide signals that are higher than noise in the full scan mass spectrometer and ignores low-abundance peptides. Thus it fails to obtain high-quality MS/MS spectra for all peptides present in the sample [2, 4].

Selected reaction monitoring (SRM) is also known as targeted proteomics, which aims to quantify the protein of interest as accurately as possible [2]. In this method, the mass spectrometer is programmed to select a specific m/z of precursor ions for fragmentation. Fragmented product ions are selected and then directed to the detector for quantification, resulting in a trace of signal intensity versus retention time for each "transitions"(precursor-product ion pair). Several suitable transitions make up the SRM assay which used to quantify target peptides and corresponding target proteins.

This requires selection and validation of a suitable set of SRM transitions for each target precursor ion, and optimization of other SRM analysis parameters before the analysis can be applied to protein detection and quantification [5], which means more upfront investments are needed compared to discovery-based experiments [2].

Data-independent acquisition (DIA) or SWATH(which uses 25 Dalton as a scanning interval in DIA experiment) theoretically combines the advantages of DDA and SRM [6]. Instead of picking peptides for fragmentation based on signal intensity, DIA records the retention time(RT) of all peptides and all their fragment ions within the entire LC gradient, which means all data include RT, intensity and m/z of all precursor ions and fragment ions are collected. Specifically, for a specific retention time, MS1 scans the entire mass range once, records the RT and the corresponding m/z of all precursor ions, and then enters MS/MS which scans the m/z range according to the precursor ion separation window sequentially and records all fragment ion information. For example, in SWATH-MS [3] setting, the specified m/z window is usually 400-1200m/z and the precursor isolation window is set to 25 m/z. All peptides in 400-425m/z mass range will be fragmented and detected simultaneously at the first MS/MS scanning. Then, the MS/MS scanning window is increased by another isolation window to 425–450 m/z, and this process is repeated 32 times, gradually completing the entire mass range, this forms a cycle time [6].

Data acquired in DIA mode is continuous in time and fragment ion intensities, which allows complete mass spectral data with low variability. However, due to the large precursor ion separation window, peptides co-elution and co-fragmentation result in very complicated MS/MS spectrum and the loss of the direct correspondence between fragment ions and precursor ions. Thus DIA mode requires more powerful data analysis tools than the previous two methods [6].

Quantitative proteomics

Quantitative proteomics not only identifies proteins, but also measures their abundance. Accurate quantification can provide evidence for an organism's response to the internal or external environment [7]. Initially, immunological methods such as western blot (WB) and enzyme-linked immunosorbent assay (ELISA) are used to quantify proteins. But the number of proteins that can be quantified at the same time is limited [7]. Nowadays, mass spectrometry was used in proteomics researches and enables accurate quantification of hundreds of proteins [7].

Label-based quantification and label-free quantification

There are two different methods for protein quantification, i.e. label-based quantification and label-free quantification. Label-based protein quantification uses two stable "heavy" and "light" isotopes to label peptides or proteins. In mass spectrometer, the two isotopes undergo a mass shift of 3-4 Da and their peak intensities ratios are capable of indicating abundance [8]. Label-free quantitative proteomics relies on extracted ion chromatograms (XIC) intensities or spectral counting. It uses a comparison of precursor ion intensities between MS runs to quantify proteins or peptides based on peak area or peak intensity measurements after features alignment based on retention time, m/z, charge state, etc [9]. Label-based quantification is useful but limited by the number of labels that can be applied simultaneously in an experiment [7].

Relative and absolute quantification

Protein quantification can also be divided into relative and absolute quantification. Relative quantification aims to study the relative abundance of a certain protein or peptide between different biological states. In absolute quantification experiments, a labeled standard peptide with known concentration, which has the same sequence and physical properties as the sample peptide is added to the target protein. The exact amount of target protein can be measured by measuring reference peptides [7]. Both label-based and label-free quantification can be used for relative and absolute protein quantification.

Differential expression

Differential protein expression research is one of the key research directions of quantification proteomics. It aims to study the abundance and expression of proteins in different biological states to identify biomarkers of disease and treatment response, elucidate biological pathways, and identify and validate protein drug targets.

Analysis of protein differential expression typically involves normalization of quantitative matrices and comparisons of statistical differences in the abundance of proteins. Normalization of the quantitative matrix is necessary to scale the data to a range that makes the samples more comparable, thereby eliminating technical issues caused by repeated experimental designs and instrumentation bias to ensure the reliability of downstream analysis [10, 11].

Variance Stabilization Normalization (VSN), Linear Regression Normalization, and Local Regression Normalization are commonly used quantitative matrix normalization methods [10–12]. NormalyzerDE combines a general retention time (RT) segmentation method compatible with a broad range of global normalization methods for quantitative matrix normalization [11]. This method detects more peptides without losing precision compared to traditional methods. Downstream differential expression analysis often involves the handling of missing values and the choice of statistical methods including two-samples t test, ANOVA ect. Compared to ANOVA, NormalyzerDE achieves higher recall using the empirical Bayes Limma approach [11].

DIA quantification analysis

Data Independent Acquisition (DIA) mode enables high-throughput and highly reproducible quantification of proteomics data and has become more and more commonly used method for protein quantification [7]. It is a powerful method, but requires more data analysis than DDA and SRM modes. In DDA mode, the relationship between precursor and fragment ions is well defined. The peptide sequence that best explains a given MS2 spectrum can be found by matching fragment ions to a theoretical library generated based on a given list of proteins(spectrum-centric approach).

In DIA mode, the precursor ions separation window is generally large, resulting in co-elution and co-fragmentation of precursor ions within the same retention time, and the direct relationship between precursor and fragment ions is lost. Therefore, DIA produces convoluted and highly co-fragmented spectra that require more complex analytical tools to make sense of the produced signal.

Recently, many DIA quantitative analysis software have been published. For example, DIA-Umpire deconvolves the DIA data and redistributes the fragment ion in the MS/MS spectrum as a pseudo-DDA spectrum through a certain algorithm, and then performs a spectrum-centric scoring to find the peptide sequence from the spectral library that best explains a given MS/MS spectrum [4]; OpenSWATH uses a peptide-centric scoring, which starts with target peptides list, aims to assign confident MS/MS sprctrum to each peptide. Regardless of the scoring method used, standard DIA quantitative analysis workflows require a reliable spectral library for chromatogram extraction, which also relies on robust chromatographic extraction tools. The extracted PSMs require further filtering, which is achieved by using powerful algorithms for peak group selection and scoring to identify mismatches. By controlling for the false discovery rate (FDR), the correct peak groups can be used to quantify protein abundance.

Chromatographic extraction

The first step in DIA quantitative data analysis is to extract the chromatographic information from the MS/MS map. In a peptide-centric scoring analysis, the algorithm first searches from a given spectral library, extracting the RT, intensities and m/z of the precursor and fragment ions and relevant chromatographic features for each peptide.

The spectral library is the basis for peptide-centric scoring analysis [3]. A spectral library usually contains a list of proteins and peptide sequences digested under specific conditions, the retention time of the peptide, the empirically based precursor ion of the peptide with strong signal and its m/z, the fragmentation of the precursor ion under specific conditions, the intensities of the resulting fragment ions and their m/z [3]. It also contains the information of a group of reference peptides which is added into samples for retention time normalization. Due to the fact that the retention times of peptides varies while the settings of the LC setup is different, retention time of this experiment can be normalized and calibrated by aligning its retention time back to the library [13]. There are several methods can be used to generate spectral libraries. It can

be derived from previously collected DDA data. In SWATH-MS experiments, a "peptide query parameter" (PQP), which stores prior knowledge of the mass spectrometry and chromatography of target peptides in an empirically based spectral library [3] is used as library. It is also possible to directly deconvolve DIA data to obtain pseudo MS/MS spectra to generate spectral library(*DIA-Umpire* [4]). Searle et al. [14] generated a chromatographic library at the expense of adding 6 gas phase fractionation injections to the narrow-window DIA run, which results more sensitive and reproducible than generating a spectral library from DDA mode.

Chromatographic extraction also relies on a powerful library search engine. An increasing number of library search engines have been published for DIA data analysis. All of these search engines introduce mismatches, so downstream analysis mainly focuses on identifying mismatches generated by search engines.

Peak group scoring

Several "peak groups" can be identified in a transition group record [15], but usually not all peak groups can represents the peptide of interest. Selecting the correct peak for each precursor ion ensures accurate quantification, so it is one of the most important steps in a DIA quantitative analysis. Peak group scoring is generally performed by training machine learning models to score target peaks and decoy peaks based on peak features.

Several algorithms and models have been published in recent years to select optimal peaks for precursors. mProphet [15] uses a semi-supervised learning to combine sub-scores such as temporal concurrency and the shape of peaks in peak groups, and the relative intensities of transition traces to obtain a discriminant score for each peak to maximize the separation between true and false peak groups([15]). Peak groups with the highest discriminant scores were used for protein quantification. DIA-NN [16] trains a linear classifier iteratively according to 73 peak features such as fragment ion co-elution, similarity between observed and reference spectra library etc. to calculate discriminant scores. Then it trains a small ensemble of neural network to calculate q values based on discriminant scores for each peak to control FDR.

Target-decoy strategy is commonly used to identify PSMs in DIA data analysis. In general, mismatches are definitely present due to the fact that not all peptides in a sample will be present in the search space but still assigned to a spectrum, and all MS/MS spectrum including those from the background are usually given peptide matches [17]. In experiments using precursor intensity for quantification, peak groups and their fragment ions are used to calculate precursor intensity. If the intensity of a precursor ion is calculated by incorrect peak group, it can lead to a inaccurate quantification of this peptide. So, identifying and filtering mismatches ensures accurate downstream quantification.

In DIA experiments, it is not practical to examine all PSMs manually. A commonly used strategy is the target-decoy strategy [17]. The goal of target-decoy strategy is to add a naturally nonexistent, necessarily incorrect "decoy" sequences to the search space, which, when deemed correct, can represent those false matches in the search results. These decoy sequences are similar with incorrect target sequences in length, amino acid composition and scores calculated by search engine [17], which indicates that decoy and mismatches have similar MS/MS spectra. Through this strategy, a classifier can be trained from multiple features and can be used to identify mismatches from the target. The target-decoy strategy is simple and powerful, it is applicable to data generated by any search engine [17]. Thus, it has become the main method used by many software to select peak groups.

False discovery rate(FDR) is a concept in statistics that is a way of conceptualizing the false positive rate (i.e. type I error in null hypothesis testing) when making multiple comparisons. The FDR is the expected ratio of the number of false positive classifications in the total number of positive classifications [18]. It can be calculated from the confusion matrix (detailed in Fig 2) by the following equation:

$$FDR = FP/(FP + TP) \tag{1}$$

Where FP refers to false positive, TP refers to true positive.

FDR control procedures use a given q value to control FDR due to the fact that FDR is the expectation value of q. Q value can be calculated by decision score [19]. A binary classifier calculates the possibility(or decision score) of data being classified into a classification (Supplementary figure 1). According to a given threshold of decision score, data points can be classified into a specific classification (target or decoy) and resulting in a confusion matrix. At a given possibility threshold, q values can be calculated by the following equation [19]:

$$Q = B/(B + A) \tag{2}$$

Where B is the number of false discoveries and A is the number of true discoveries.

Most of the DIA quantification software are performed using the target-decoy strategy followed by FDR control. For example, EncyclopeDIA [7] search results were filtered to a 1% FDR on peptide-level using Percolator [20]. Proteins are then parsimoniously allocated to protein groups and filtered on 1% FDR at protein-level [21]. OpenSWATH [22] search results were filtered to a 1% FDR on peptide-level using pyProphet [23] if peptide and protein inference in the global context is conducted [22].

Calculation of proteins abundance

In label-free experiments, the abundance of proteins can be calculated by spectral counting or MS1 intensity calculations [9]. Spectral counting works on the principle that peptides with higher abundance produce more abundant MS/MS spectra after fragmentation, so the peptide counts assigned to a protein are proportional to the protein's abundance [9].

The principle of calculating the intensity of MS1 (or precursor) is to quantify peptides by summarizing the intensities of fragment ions extracted from the MS/MS spectrum. Obviously, the population of fragment ions used to calculate precursor intensity will directly affect the final quantitative results. The most commonly used method in the past is the TopN method [24], which is the summation of the intensities of N fragment ions with the highest intensity [25], or directly sums the intensities of all fragment ions. Software, such as OpenSWATH and EncyclopeDIA, rank the importance of these ions by their overall intensity and then sum them all up to provide quantitative information for a peak group that corresponds to a peptides precursor ion. Although this works for the most part, there have been increasing number of studies that have found that performing statistical analysis of proteins on the fragment ion level provides more accurate quantitative information and more precise differential expression data [24, 26].

However, these approaches tend to ignore the fact that multiple ions can contribute to the overall intensity of a detected fragment ion, or they claim that this contribution is insignificant. In complex samples, these interferences can be quite common, and effectively skew quantitative accuracy by assuming the most intense ions provide the best quantitative information, when in fact the high intensity of that ion species is the result of multiple fragment ions interfering with each other. Approaches have been developed to correct fragment intensities based on levels of identified interference [23] or

to identify interfered fragments for consideration in downstream processing [16]. It has been shown that by using multiple sub scores, including fragment ion interference, it is possible to identify and rank fragment ions and select optimal subsets of these fragment ions to significantly improve the quantitative accuracy of peptide identification (11-25%) versus the TopN strategy [25].

Machine learning (ML)

Machine learning is a discipline and method that uses computers as tools to simulate the way humans learn in real time. Machine learning can learn from data or previous experience to improve the performance of specific algorithms in artificial intelligence, or to optimize the performance metrics of computer programs. The goal of machine learning is to train a machine learning model to do a task well. The performance of machine learning model on this task can be measured and the performance can be improved by learning from experience [27]. Specifically, machine learning learns from a training set with the aim of building a model based on sample data in order to make predictions or decisions without being explicitly programmed.

Machine learning models are trained by minimizing the loss function and adjusting the weights accordingly. The loss function represents the difference between the predictions of the model being trained and the actual problem instance [28]. Therefore loss functions can also be used to evaluate how well a particular algorithm models a given data. If the prediction deviates too much from the actual result, the loss function will produce very large numbers, so the goal of machine learning is to let the loss function learn to use some optimization function to reduce the error of the prediction.

Machine learning problems can be divided into classification problems and regression problems. In regression problems, machine learning learns from data to predict accurate values (Fig 1a). Whereas in classification tasks, machine learning learns the data and classifies it into specific categories (Fig 1b). Correspondingly, classification models are trained by finding optimal parameters to minimize the classification loss function, and regression models are trained by minimizing the regression loss function.

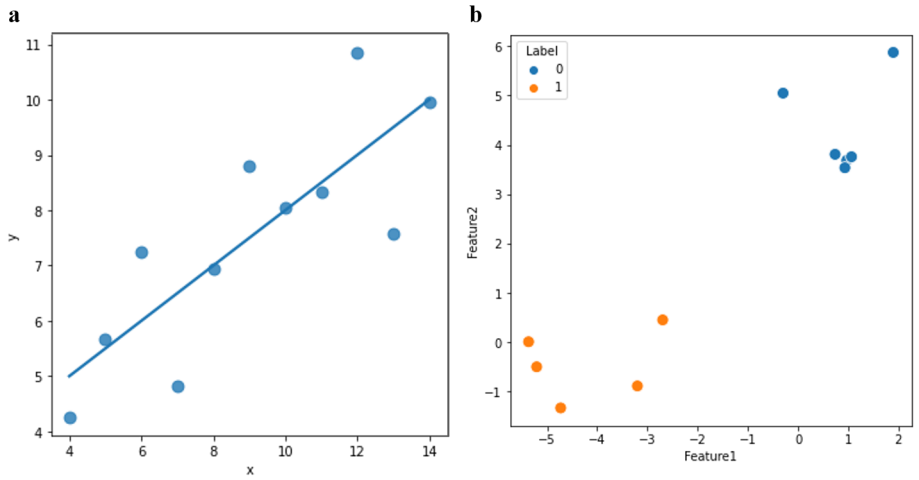

Fig 1. The tasks of machine learning are divided into classification tasks and regression tasks. a) Regression tasks(a univariate regression). b) Classification tasks(a binary classification).

Binary classification [29]

Binary classification is commonly used in DIA data analysis because of target-decoy strategy. Binary classification refers to the classification problem where the label has only two classifications while multi-class classification refers to the classification problem where data are labeled in more than two classes. The task of binary classification model is to learn the characteristics of the two types of labels on each feature, and classify the data into one of the two categories. It can be used to predict whether email is spam or whether fragment ions and peak groups are targets.

Algorithms that are suitable for classifying data set are called classification algorithms. Popular algorithms for binary classification includes logistic regression, k-nearest neighbors, decision trees, support vector machines (SVM), and naive bayes. Among others, logistic regression and SVM algorithms are specifically designed for binary classification.

Stochastic gradient descent (SGD) [30]

SGD also known as incremental gradient descent, is an iterative method for optimizing differentiable objective functions. It is not a machine learning model, but an optimization technique used to train a machine learning model. This is a simple but very effective method and can be easily implemented [30]. Specifically, it uses gradient information to minimize the objective function by continuously iteratively adjusting the parameters to find a suitable value. It does not search for the parameter that minimizes the loss function on all training data, but randomly optimizes the loss function on a certain piece of training data in each round of iteration, so that the update speed of each round of parameters is greatly accelerated, so it is the best method to train a model on the large data set. It is commonly used for fitting linear classifiers and regressors under convex loss functions, such as (linear) support vector machines and logistic regression [30].

Model evaluation [31]

Model evaluation is to measure how a machine learning model good at a task. Take binary classifier as an example, a binary classifier calculates a probability of being classified into class 1 (also known as the positive class) for each given data point. The probability then is used to determine the predicted class based on a probability threshold(or decision score threshold). The performance of the classification can then be assessed by comparing the predicted values with the actual values and summarizing the results in a 2x2 matrix (i.e. confusion matrix) showed in Fig 2. A data point that is actually a negative class (0) and is predicted to be a negative class (0) is called true negative (TN); a data point that is actually a negative class (0) and is predicted to be a positive class (1) is called false positive (FP); a data point that is actually positive (1) and is predicted to be positive (1) is called true positive (TP); a data point that is actually positive (1) and is predicted to be negative (0) is called false negative (FN). The final result for each data point can only be one of four. And the number of 4 results present in the evaluated data set forms the confusion matrix. Confusion matrix varies while varying the threshold used to assign observations to a given class.

Precision value, recall value and ROC-AUC value calculated by confusion matrix are frequently used to evaluate the performance of a model. Recall score, (also called true positive rate or sensitivity) as marked in green in Fig 2, refers to the percentage of the number of positive data points that are correctly predicted as positives within the number of all actual positives. Precision score, as marked in red in Fig 2, refers to the percentage of the number of positive data points that are correctly predicted as positives out of the number of all predicted positives.

The ROC-AUC score is also one of the main metrics for evaluating the performance of a classification model [29]. The ROC-AUC score is obtained by calculating the area under the receiver operating characteristic(ROC) curve, which represents the degree of separability/distinction or admixture/intersection between the predictions of two classes and summarizes the performance of a classifier over all possible probability thresholds. Specifically, new confusion matrix is generated while decision score threshold changing. The ROC curve then can be generated by taking true positive rate(TPR, also know as recall, as marked in green in Fig 2) as the y axis and false positive rate (FPR, as marked in yellow in Fig 2) as the x axis. The ROC-AUC score then can be calculated. The higher the ROC-AUC score, the higher the discrimination between the two categories and the better the performance of the model.

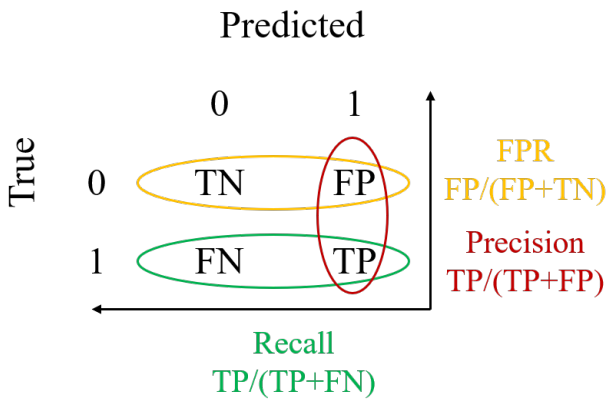

Fig 2. Confusion matrix and measurements of models.

Precision/recall trade-off

Precision/recall trade-off is defined from the fact that high precision and high recall generally cannot be satisfied simultaneously: increasing precision reduces recall and vice versa(Fig 3). It is a strategy to improve the performance of machine learning models. The goal of this strategy is to find a possibility threshold that will provide the machine learning model with target precision and recall score to ensure the performance of model in classifying the data into one of four results. As can be seen from Fig 3 marked with red line, when the precision is 0.9, there is a certain recall value and a threshold for the decision score. And with this decision score threshold, at least 90 precision score can be archived by the classifier.

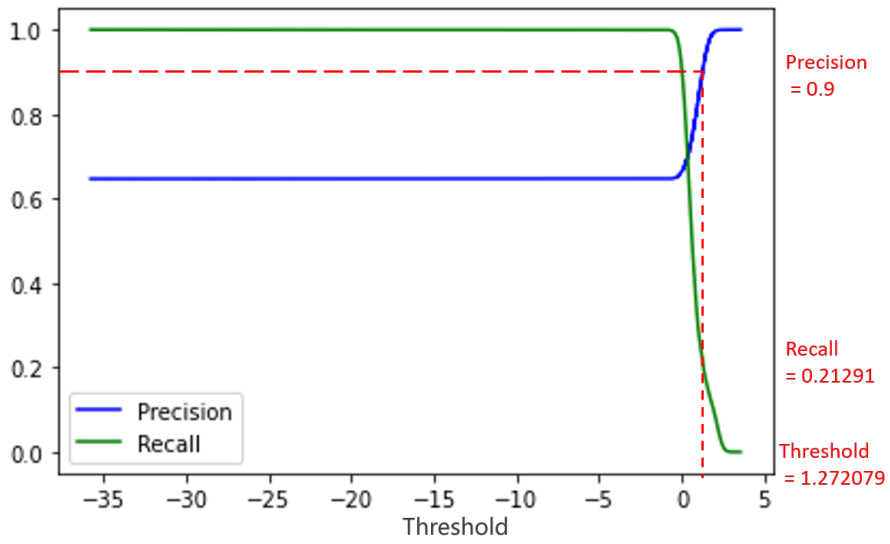

Fig 3. The precision-recall trade-off of a classifier.

Classifiers usually use 0 as the threshold. When a threshold higher than 0 is used, higher precision and lower recall are obtained. Different classification tasks have different purposes and it can be achieved by using precision/recall trade-off. In the target-decoy strategy, it is important to ensure that the probability of true 0 (decoy) being predicted as 1 (target) is as small as possible, so the purpose is usually to guarantee a high precision instead of high recall. However, in some models such as fraud detection models, it is not expected for a true 1 to be predicted as a 0, so the goal is usually to ensure that the recall score is as high as possible.

Research aim

This project aims to investigate the possibility to streamline fragment ion ranking and automate the optimal quantification of peptides using a subset of the fragment ion population for each precursor. We aim to use algorithmic approaches and machine learning to determine fragment ion ranks within a peak group, and then combining them in an optimal way to provide quantitative information at the peptide levels. The machine learning involved is a classification problem, where negative target labels are derived from the fragment ions extracted from the decoy entries in the spectral library. Algorithms will also be implemented to calculate the features that will be used as input for the classifier, as well as multiple methods to then combine the selected fragment ions in an optimal way to provide accurate peptide level quantification. The algorithms will all be implemented in an existing Python package developed by the group, so that they can be easily accessible for analysis for particular data set. Benchmarking will be done using complex samples with specific sets of peptides spiked-in at known concentrations so that quantitative accuracy can be precisely measured based on a ground truth.

Materials

All files involved in this project are stored in the group Linux server, where most of the software is installed. But some software such as NormalyzerDE was used as R package in Rstudio of 64-bit Microsoft Windows 10 system on the local computer. Code and data is available at https://github.com/Lina0125/QuantifyAtFragmentLevelWithClassifiers.

Yeast benchmark sample preparation

To obtain the benchmark proteomic data, we divided the experiments into two groups, each containing 10 samples. Spike-in mouse protein are constant in all samples while yeast proteins are vary in two groups. The accurate spike-in protein concentration is showed in supplementary table 1, relatively, the concentration of the yeast protein of group2 was diluted 4-fold compared to that of group1.

Due to the fact that the settings of the chromatography system (LC) affect the elution times of all peptides, specific LC settings result in RT data for peptides that are specific to a single experiment. A group of iRT peptides used for retention time calibration is added to the sample during the sample preparation stage, and is used to perform regression against the experienced library to correct the retention time of this experiment. There was only a slight difference in the concentration of iRT peptides between the two groups (Supplementary table 1). The iRT intensity of the two groups in the differential expression result should be the same theoretically.

LC-MS DIA data collection

All peptide analyses were performed on a Q Exactive HF-X mass spectrometer (Thermo Fisher Scientific) connected to an EASY-nLC 1200 ultrahigh-performance liquid chromatography system (Thermo Fisher Scientific). Peptides were trapped on the precolumn (PepMap100 C18 3 µm; 75 µm × 2 cm, Thermo Fisher Scientific) and separated on an EASY-Spray column (ES803, column temperature 45 °C, Thermo Fisher Scientific). Equilibrations of columns and sample loading were performed per manufacturer's guidelines. Solvent A was 0.1% formic acid, and solvent B (0.1% formic acid, 80% acetonitrile) was used to run a linear gradient from 5 to 38% over 120 min at a flow rate of 350 nL/min. The mass range for MS1 was 350–1650 m/z with a resolution of 120,000 and a resolution of 30,000 for MS/MS with stepped normalized collision energies (NCE) of 25.5, 27, and 30. The data-independent acquisition (DIA) method was derived from Bruderer et al. [32]. The 44 variably sized MS/MS windows were 350–371, 370–387, 386–403, 402–416, 415–427, 426–439, 438–451, 450–462, 461–472, 471–483, 482–494, 493–505, 504–515, 514–525, 524–537, 536–548, 547–557, 556–568, 567–580, 579–591, 590–603, 602–614, 613–626, 625–638, 637–651, 650–664, 663–677, 676–690, 689–704, 703–719, 718–735, 734–753, 752–771, 770–790, 789–811, 810–832, 831–857, 856–884, 883–916, 915–955, 954–997, 996–1057, 1056–1135 and 1134–1650 m/z,resulting in a total cycle time of ∼3.3 s and 6–8 sampling points per chromatographic peak on average.

Raw data converted to mzML

All DIA data analysis software take open file format such as mzml [33] files as input. Thermo raw data from mass spectrometers can be converted to mzml files by ThermoRawFileParser [34] or MSconverter in ProteoWizard [35].

Spectra library

The spectra library contains precursor and fragment ion m/z values, relative fragment ion intensities of transitions and normalized peptide retention times. Decoys are added to the spectra library for later classification and error rate estimation. Spectra library for yeast benckmark data is already available in this project and was generated using MSFragger then transformed to OpenSWATH available peptide query parameters (PQPs), i.e. the pqp file format. The PQP is a parameter required for peptide-centric scoring, which stores the following information: the optimal peptides for the protein,

and the elution times of those peptides under specific LC settings; Several fragment ions with the strongest signal generated by the fragmentation of precursor ions under specific collision conditions; the charge states of the precursor and fragment ions and the relative ion intensities of all selected fragments [22]. The spectral library also contains decoy proteins that can be generated in OpenSWATH by OpenSwathDecoyGenerator for later classification and error rate estimation.

iRT library

A kit of 8 reference peptides have been added into samples. The iRT library contains the PQP of these peptides is used to calibrate the retention times for this experiment. Specifically, Regression alignments of retention times for the reference peptides and iRT library were performed from a single run to obtain regression equations that were mapped back to the spectral library to normalize and calibrate retention times.

OpenSWATH

OpenSWATH is a peptide-centric search software which performs targeted chromatographic extraction. Its components are very flexible, and can be compatible with other software only by converting data formats, and allows users to add their own functions to form new workflows. The input files for the OpenSWATH algorithm contains the spectral data mzml files, the spectral library, the iRT library and the swath window file which contains the MS window settings for this experiment. OpenSWATH is used to conduct retention-time alignment, chromatographic extraction and peak-group scoring in this project and it is used as a singularity container [36].

pyProphet [37]

pyprophet is an implement software of mProphet [15]. It uses a number of different subscores such as the pearson correlation between each pair of corrected traces, performs semi-supervised learning on decoy and target peaks, merges the sub-scores into a final score, selects the best peak in each chromatogram, and estimates FDR [23]. The pyProphet software used in this project is directly installed in the virtual environment of the Linux server and was installed using the following command:

```
pip install pyprophet
```

TRIC

TRansfer of Identification Confidence (TRIC) uses cross-run alignment and retention time correction to further align and correct stacking error conditions from multiple OpenSWATH and pyProphet runs. Only the feature_alignment.py script is used in this project, which is directly placed in the site package in the virtual environment of the Linux server. It was installed using the following commands:

```
pip install numpy
pip install pymzml==0.7.8
pip install Biopython
pip install msproteomicstools
```

Snakemake [38]

The Snakemake workflow management system is described based on the Python language, which allows the integration of many different software and languages in a

workflow, which is repeatable and extensible, and can be easily deployed to any environments. The Snakemake version used in this project is 6.13.0.

Singularity container

Singularity virtualizes and containerizes systems, allowing applications to be moved from one system to another [36]. The OpenSwathWorkflow in this project is run through Singularity, and the program is currently available in the group.

Python

Python is a programming language that is powerful, fast, and portable. The main steps of scoring fragment ions and filtering by user specified methods is all written into a Python package by using version 3.8.10.

NormalyzerDE [11]

NormalyzerDE is a LC-MS-based protein quantification analysis software for normalization and differential expression analysis. It is used to normalize quantitative matrix and conduct differential expression analysis in this project. We use the NormalyzerDE R package in Windows system 10 Rstudio 64-bit with R version 4.0.1 on the local machine. Installation was performed using BiocManager:

```
install.packages("BiocManager")
BiocManager::install("NormalyzerDE")
```

Methods

As shown in Fig 4, we combine the OpenSWATH workflow with machine learning classifiers and an algorithm to score fragment ions. The experimental data were already converted into mzML format. At the beginning of our work, OpenSWATH is used to obtain sqlite3 tables which store PSMs and their information. We use the merged process of pyProphet and TRIC as a generalized workflow to compare with our workflow. In the optimized workflow, we first train several different classifiers using the output of OpenSWATH, they are "initial", "precision_90", "qvalue_cutoff" and "lnl" classifiers. These classifiers are then used on the pyProphet scoring data to obtain discriminant scores for fragment ions, which are used to calculate q values for FDR control at the fragment ion level. Finally, quantitative matrices are obtained by different precursor intensity calculation methods (TopN, mean, median, sum).

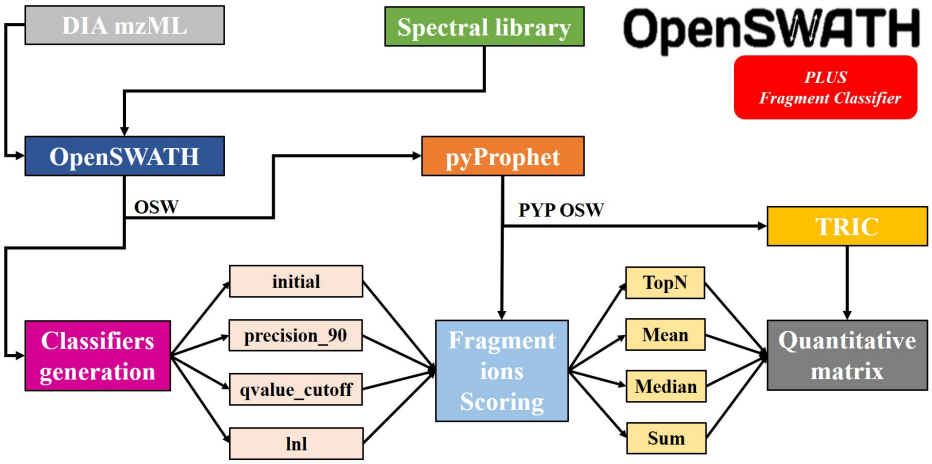

Fig 4. Integrate fragments ion classifier into OpenSWATH workflow [22].

Generalized Workflow and parameters

We use OpenSwath Workflow as a generalized method with 3 components as shown in Fig 5; In this workflow, OpenSwath is used for targeted extraction of chromatogram; pyProphet is used for statistical scoring; TRIC is used for alignment of multiple runs to generate quantitative matrix.

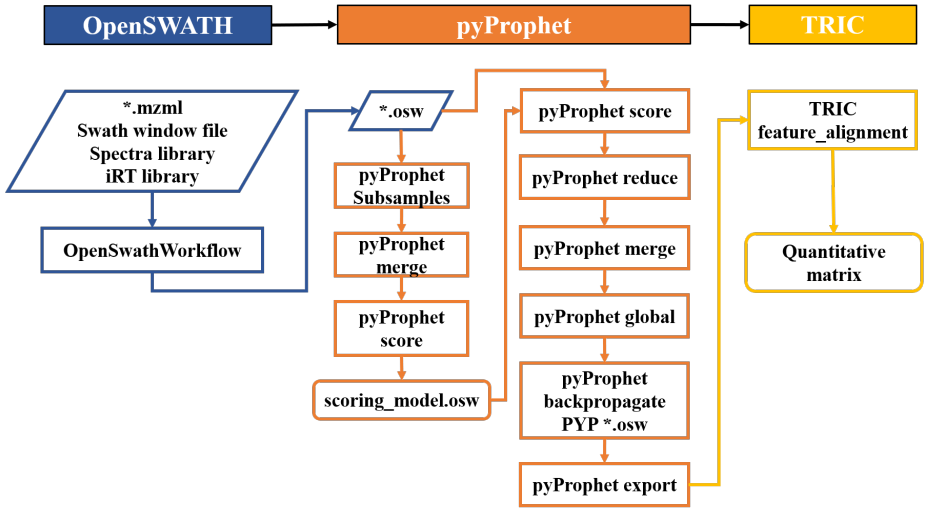

Fig 5. Detailed steps in generalized workflow.

OpenSWATH is used to extract chromatogram. As shown in Fig 5 marked with blue, OpenSWATH takes mzml files, spectra library, iRT library and swath window file as inputs. Firstly, it calibrates retention times in each run by aligning them against a previously determined normalized retention time space generated by mapping iRT library in a run back to the iRT library. Then it searches from peptide lists in spectral library, with the instructions of search window in swath window file, it extracts MS/MS map for each peptide according to the m/z and retention time information from the spectra library to generate integrated fragmentation counts versus retention-time data.

Finally, OpenSWATH conducts peak groups scoring algorithm by scoring spectrum features using multiple, orthogonal scores such as elution profiles of the fragment ions etc. The extracted chromatogram data and scores in every run are stored in the output osw files. The intact parameters of OpenSWATH used are:
OpenSwathWorkflow
-in
-tr
-tr_irt
-out_osw
-out_chrom
-swath_windows_file
-threads 50
-enable_ms1 true
-enable_ipf true
-Scoring:Scores:use_uis_scores
-Scoring:Scores:use_total_mi_score
-Scoring:stop_report_after_feature 5
-Scoring:TransitionGroupPicker:compute_peak_quality
-min_rsq 0.07
-batchSize 10000

pyProphet [37] is used to score and choose peak groups and is conducted following OpenSwathWorkflow documentation(Fig 5 marked with orange). The pipeline is run-specific context and can be separated to 2 parts: model generation and model application and global output.

In model generation step, firstly, "pyprophet subsamples" subsample all runs. Then "pyprophet merge" generates a subsampled classifier that learns faster. Then the classifier is learned at "MS1/MS2" level using "pyprophet score" and the results are stored in scoring_model.osw.

In model application and global output step, "pyprophet score" applies the classifier generated in the first step to all specific runs in parallel. Then "pyprophet reduce" extracts relevant data for global scoring to generate small files (*.oswr). Next, "pyprophet merge" merges the files, and performs global peptide ("pyprophet peptide") and protein level ("pyprophet protein") FDR control. "pyprophet backpropagate" backpropagates global statistics to individual runs. Outputs from backpropagates(PYP osw files) are also used as the input of optimized workflow(Fig 6)b. Finally, "pyprophet export" exports results with confidence scores at the peptide query level (run-specific context).

TRIC [39] is used to align peaks within runs. As shown in Fig 5 marked with yellow) . Results from pyProphet that with confidence scores on peptide query level then aligned in TRIC. Peak groups filtered at the 1% FDR level and stored in the final quantitative matrix. The intact command used is as follow:
python feature_alignment.py
–in
–out
–out_matrix
–max_rt_diff 30
–mst:useRTCorrection False
–fdr_cutoff 0.01
–max_fdr_quality 0.01

The optimized workflow

In order to select the most suitable fragment ions for each peak group, we use machine learning to learn 12 fragment ions features such as the intensity score compared to intensity in library of each transition(VAR_INTENSITY_SCORE) and the mass deviation score of the fragment ions throughout the retention time(VAR_MASSDEV_SCORE) etc. to discriminate true and false fragment ions. This is achieved by adding decoys to the library searching space to provide negative labels and its similar spectrum maps with positive targets for classifier training. The resulting models are then used to generate a discriminate score. Discriminate scores are then used to calculate q-values which can be used to control the FDR at the fragment ion level.

Models generation

As shown in Fig 6a, the data for building the classifier comes from the output of OpenSWATH. OpenSWATH extracts the features of transition groups from the chromatogram and stores them in the osw tables. The 12 sub-scores are used as classifying features, and the label is the target and the decoy added to the search space. Specifically, we first use sqlite3 in python to merge SQL tables. In model training process, data is firstly separated to 80% training set and 20% test set("hold-out cross-validation" training method), then models are trained using SGD method. Specifically, we fit a logistic regression model to the training data and optimize the model via SGD, which is implemented using the SGD Classifier(loss="log_loss") in scikit-learn [40]. This model is called the "initial" model and is used as a reference classifier. We then apply this classifier to the data and use the result as a reference. Three new classifiers are eventually generated by using three different methods to improve the performance of the "initial" classifier.

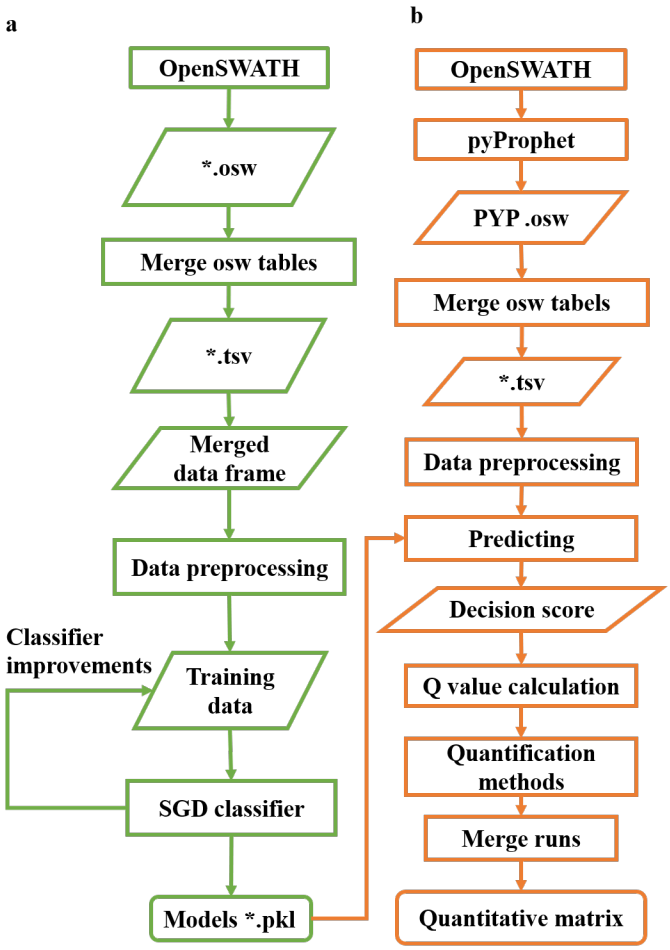

Fig 6. The steps of optimized workflow. a)Models generation; b) Application of the optimized workflow.

Model improvements are mainly achieved by ensuring a high precision score and dealing with noisy labels. Specifically, we used three different strategies: 1) Control the precision to at least 0.9 through the precision/recall trade-off strategy, then train a new classifier on the filtered reliable training set by the same method, this classifier is called "precision_90". 2) Generate another new training set by removing the noisy labels (false targets) in the training set through the FDR control strategy. Specifically, the q value was calculated using the gscore python package (which is a implementation of the QVALITY algorithm [41]) to control the FDR at the 1% level and generate a new training set. Through the same process as "initial" classifier, it trains a new classifier called "qvalue_cutoff". Gscore algorithm is also used for downstream FDR control for all optimized workflow. 3) Finally, we use confident learning [42] to enhance the initial classifier. It is implemented using the CleanLab [43] package. Specifically, it trains a robust version of the initial model that performs better with noisy labels. It improves the model without removing any training indices, this classifier is called "lnl"(LearnWithNoisyLabel).

Due to the deletion of part of the training set, the number of two categories is uneven In "precision_90" and "qvalue_cutoff" classifiers. The target/decoy ratio of "precision_90" classifier is 9:1, and the target/decoy ratio of "qvalue_cutoff" classifier is

1:5 (Supplementary table 2). Thus class weight="balanced" parameter is used in SGD classifier to balance the weight of the class during model fitting. It uses weighted classification costs to alter the behavior of the learned classifier so that it gives more weight to points in smaller classes and less weight to points in larger classes.

FDR control at fragment ion level

The entire optimized workflow is shown in Fig 6b. In order to perform FDR control at the protein and peptide levels, we use the pyProphet scoring data instead of the OpenSWATH output data as the input to the classifiers. So OpenSWATH and pyProphet is conducted firstly in our workflow. In the next steps, each run is processed individually, performing scoring, FDR control at fragment ion level, and precursor intensity calculation. Finally each run is combined into a quantitative matrix.

Specifically, in each individual run, the resulting models first classifies all fragment ions and gives a discriminant score, this discriminant score is then used to compute a q value (via the gscore algorithm) for subsequent use in FDR control at the fragment ion level. FDR control and filtering were first processed at the protein and peptide level, which was achieved by aligning each run against a list of proteins and peptides in a global model generated by pyProphet. We then perform FDR control at the fragment ion level to filter fragment ions that were misassigned to the precursor ions in the peptide list.

Precursor quantification

Finally, we calculate precursor intensities of the target peptides by four different methods. 1) TopN strategy, that is, select top N optimal fragment ions and sum its intensities; 2) Mean strategy, calculate mean intensities of all reliable fragment ions; 3) Median strategy, calculate median intensities of all reliable fragment ions; 4) Sum strategy, sums the intensities of all reliable fragment ions.

All runs are processed individually and then combined into a quantitative matrix. In order to avoid data redundancy, we removed the protein with intensity NaN generated by aligning back to the global model in all samples because it was not expressed in any samples.

Differential expression analysis

Protein quantification matrix normalization and differential expression analysis were performed in R studio on a local computer using the NormalyzerDE R package. In addition to implement the retention time segmentation method, loess regression (using the "performCyclicLoessNormalization" parameter) was also employed, merged by the mean normalization method. Data set evaluation and normalization quality assessment and differential expression analysis were performed using empirical bayesian limma method. The log2 fold change value, statistic p value and adjusted p value of the two groups at the protein level are finally reported.

Process of multi-species proteomics data

In order to investigate whether the classifiers generated in the yeast benchmark experiments can also be used for DIA data set containing multiple species. We used the optimized workflow on low-fold change experimental data from a high-complexity data set generated by Bruderer et al [32], which contained a total of four species (H. sapiens, C. elegans, S. cerevisiae, E. coli). Data is downloaded directly from PRIDE with the data set identifier PXD005573. All process of this data set is the same as the yeast

benchmark data, except that the fragment ion classifier is already derived from the yeast benchmark data. We also generated sample specific classifiers for this data set, with all processing steps as same as for the yeast benchmark data.

Code availability

The code for this project is available in https://github.com/Lina0125/QuantifyAtFragmentLevelWithClassifiers. It contains two analysis pipelines (generalized pipeline and optimized pipeline) written in Snakemake and a portable python package which can be used for quantifying DIA data at fragment ions level. The generalized pipeline uses a combination of OpenSWATH, pyProphet and TRIC to automate the routine processing of DIA data quantitative analysis. The optimized pipeline uses a combination of OpenSWATH, pyProphet, and the python package for scoring and filtering fragment ions. It further controls FDR at the fragment ion level and allows fragment ions to be combined using user-selectable methods, which enables quantification of DIA data at the fragment ion level in an automated, efficient and user-friendly manner.

Results

Generated classifiers

The trained classifier consists of an initial classifier and three classifiers trained using three different improving methods. To gain insight into how each model differs in its predictions and performance, we used Shapley values [44] to explain how every input feature contributed to the output decision among all features in every classifier. In the "initial" classifier, the mutual information score is the most important feature while the mass deviation score is the least important feature on average. If a data point has the higher mutual information score, it is more likely to be predicted as a target. On the contrary, the higher intensity score leads to higher possibility to be predicted as a decoy(Supplementary figure 2a). In "precision_90" classifier, the mutual information score is the most important, however, fragment ion intensity ratio is the least important feature(Supplementary figure 2b); In "qvalue_cutoff" classifier, the shape correlation between peaks in a peak group has most contribution to model output(Supplementary figure 2c). Among the three classifiers, the higher peak shape correlation score leads to higher possibility to be predicted as a target while higher isotope overlap score leads to higher possibility to be predicted as a decoy (Supplementary figure 2).

Before classifiers are used in every run, we evaluated the precision score of every trained classifier on the same test data set. Result shows that improved models all have higher precision score than the reference ("initial") classifier. The "precision_90" archived high precision score (0.977) as it is expected to be. "qvalue_cutoff" classifier has the highest precision score (0.980) on test data set (Supplementary table 3).

Yeast benchmark data

The applicability of our workflow was first tested using the yeast standard benchmark data. Four trained fragment ion classifiers are combined with the q value calculation algorithm to filter and optimize fragment ions for peak groups. The filtered fragment ions then were used to calculate precursor intensity using top3, mean, median and sum strategy respectively. Each classifier is combined with the gscore algorithm and four precursor calculation methods to form 16 results, which are compared with the baseline result of the generalized workflow.

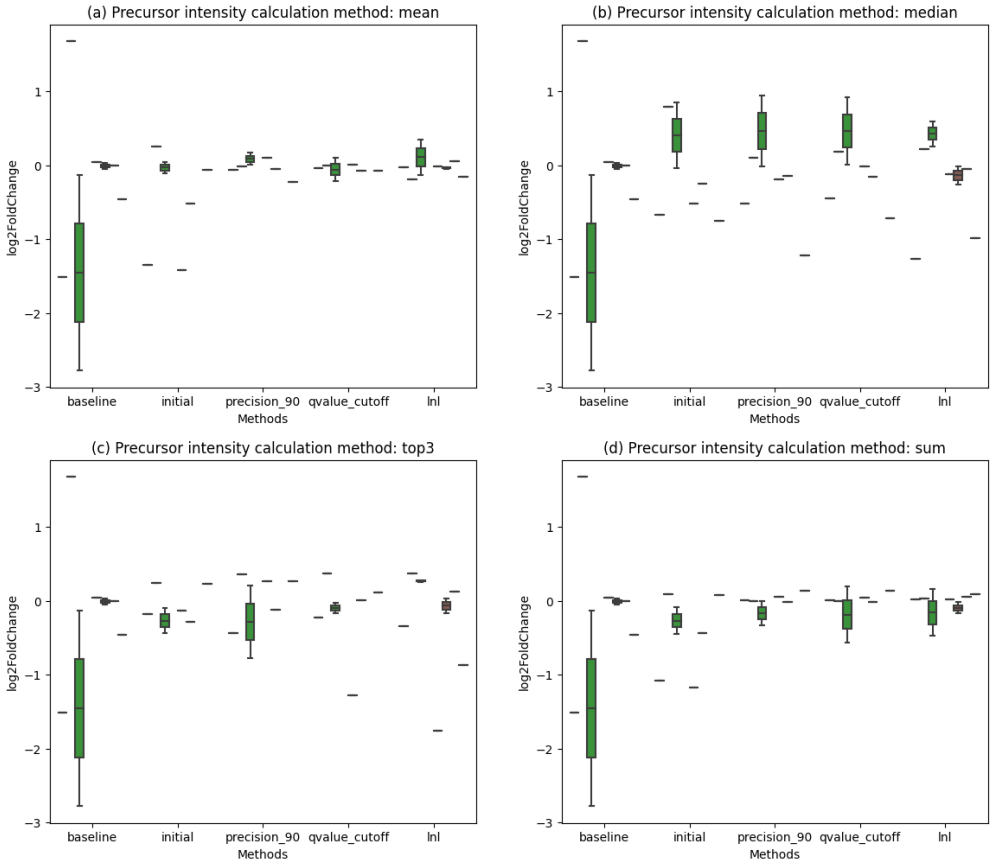

Fig 7. The differential expression result of 8 different iRT peptides. a) Mean method; b) Median method; c) Top3 method; d) Sum method.

The log2 fold changes for the 8 iRT peptides are more consistent when using optimized workflow while it is more discrete in the generalized workflow(Fig 7). Among the four calculation methods of precursor ions intensity, the mean strategy is the most accurate due to the result that the log2 fold change of the 8 iRT peptides for all machine learning methods is centered around 0(Fig 7a).

Specifically, at baseline, the mean deviation of log2 fold change values from theoretical (0) for the 8 iRTs is 0.74±1.008, higher than all classifiers combined with any quantitative method. Among all methods, the "qvalue_cutoff" classifier combined with the mean quantification strategy is the most accurate, as its mean deviation of log2 fold change values for the 8 iRTs is 0.07±0.071 (Supplementary table 4), which is the closest to the theoretical value (0).

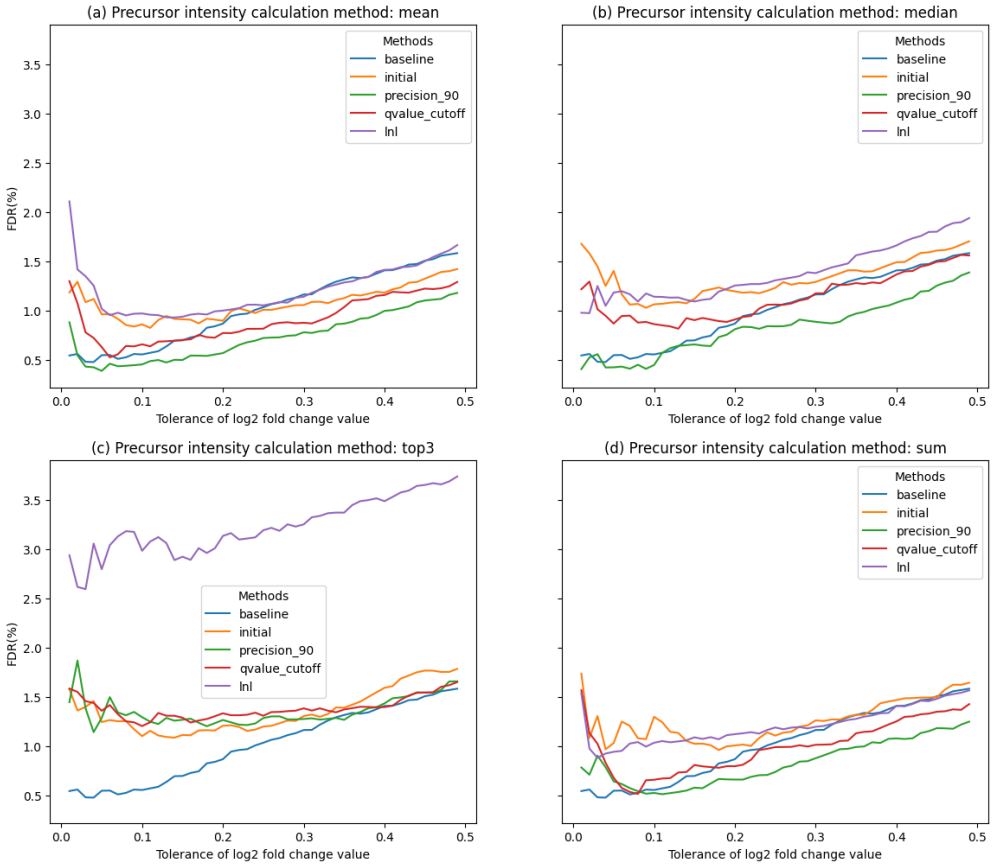

Fig 8. FDR changing curve for each method when given a ± 0.5 log2 fold change tolerance. a) Mean method; b) Median method; c) Top3 method; d) Sum method.

As shown in Fig 8, it is easy to see that the FDR values all increase while giving a ±0.5 tolerance interval of the theoretical log2 fold change value. This is because after the tolerance x is given, between 0-x - 0+x log2 fold change range, more yeast proteins are allowed to present while more mouse proteins are allowed to present in 2-x - 2+x range, resulting in an increase in FDR.

When quantifying precursor ions using mean, median, and sum methods, the FDR rise for each classifier was more gradual than baseline across the log2 fold change tolerance range(Fig 8a,b,d). Among the top3 quantification method, although the FDR of the lnl classifier is higher than that of all methods, the FDR of the rest classifiers is stable regardless of the log2 fold change tolerance(Fig 8c).

Within this tolerance range, we calculated the accuracy of every method. A mouse protein was considered correct when its adjusted p value was greater than 0.01 while a yeast protein was considered correct when its adjusted p value was less than 0.01 within ±0.5 log2 fold change tolerance range. Accuracy represents the proportion of the sum of the two values to the total number of proteins identified by each method. Results show that optimized methods have higher accuracy within ±0.5 log2 fold change tolerance range (Supplementary table 5). The mean accuracy of all optimized methods is 91.95, which is 22.46% better than the generalized method.

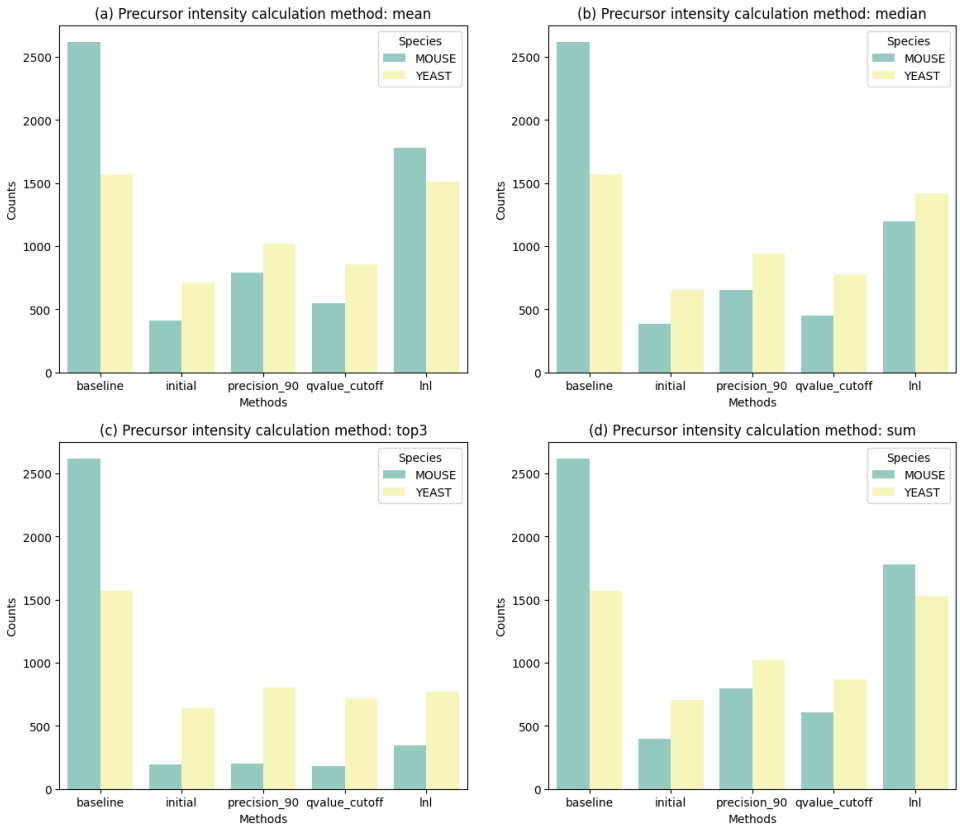

Fig 9. The number and species of protein deferentially expressed in the two groups. a) Mean method; b) Median method; c) Top3 method; d) Sum method.

When the adjusted p value of a protein is less than 0.01, we considered the protein to be differentially expressed in the two groups. Yeast proteins are expected to be differentially expressed. In the baseline, the number of mouse proteins differentially expressed in the two groups was greater than the number of proteins in yeast (Fig 9), implying a higher FDR and poorer accuracy. Among all quantification methods, the top3 strategy results in fewer mouse proteins being differentially expressed in the two groups compared to other quantification methods.

Statistically, the mean proportion of mouse proteins within differentially expressed proteins in the optimized workflow is 37.68%, which is 39.77% lower than the baseline workflow(62.56%, Supplementary table 6). This also indicates the improved accuracy of our method compare to the generalized workflow.

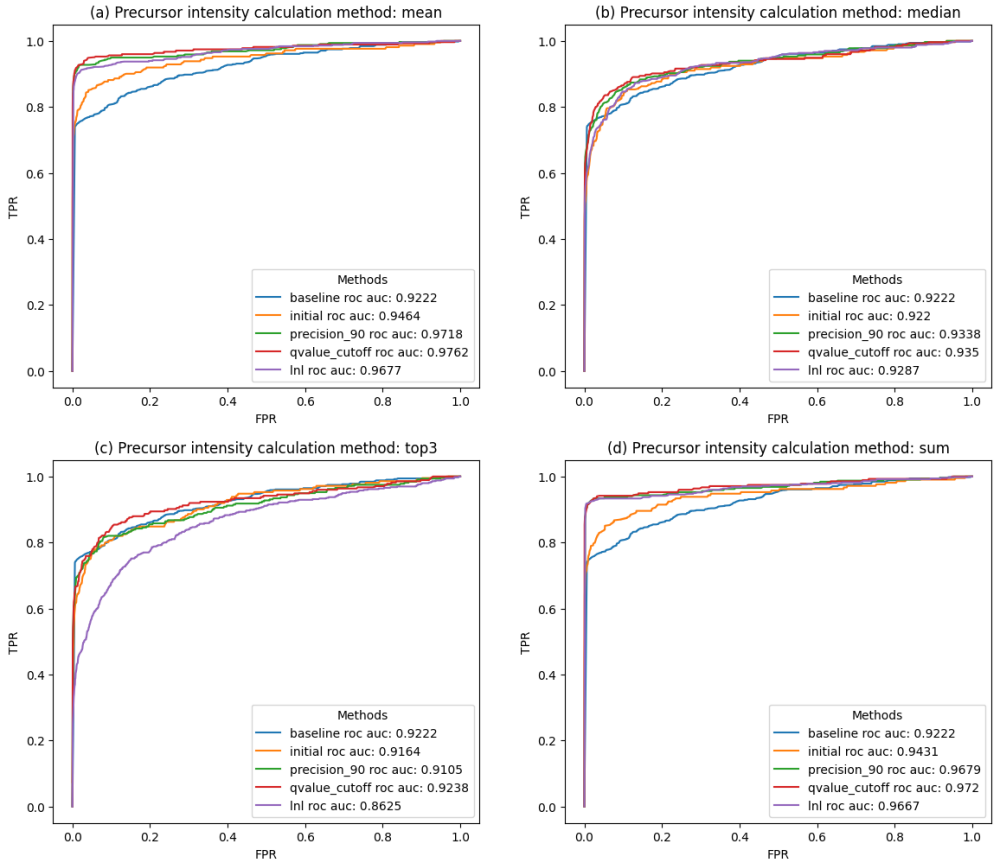

Fig 10. The ROC curve and ROC AUC score of different methods. a) Mean method; b) Median method; c) Top3 method; d) Sum method.

Using yeast proteins as class 1, mouse proteins as class 0, and 1 minus adjusted p-values as decision scores which can discriminate whether a protein is differentially expressed or not. We plotted ROC curves of FPR and TPR as a function of decision score threshold. As shown in Fig 10, FPR represents the proportion of mouse proteins among differentially expressed proteins, while TPR represents the proportion of yeast proteins among differentially expressed proteins. A ROC-AUC score of 1 indicates that all yeast proteins are differentially expressed, while an AUC score of 0 indicates that all mouse proteins are differentially expressed, i.e., the higher the ROC-AUC score, the higher the accuracy. The "qvalue_cutoff" classifier combined with the mean quantification method achieved the highest ROC AUC score of 0.9762 (Fig. 10a), indicating the highest accuracy. When using top3 quantification method, "qvalue_cutoff" achieved higher ROC AUC score than the generalized method (Fig 10c). When combine with median quantification methods, all classifiers except the "initial" have slightly higher ROC AUC than the baseline (Fig 10b) while using mean and sum quantification methods, classifiers except the "initial" classifier achieved high ROC AUC score with above 0.96 (Fig 10a,d).

Multi-species proteomics data

All classifiers have the highest ROC AUC score on average when cooperate with mean quantification method on yeast benchmark data. So mean quantification method has only been investigated on multi-species protein data. As shown in Supplementary figure

3, lines show theoretical log2 fold change value for 4 species in 2 groups(yeast:-0.263, caeel:-0.1375, human:0, ecoli:0.3785). The peak of the log2 fold change for each species in generalized workflow, "precision_90" and "lnl" classifiers are closer to the theoretical value.

We also generated sample-specific classifiers for this data set and found that the results were not different from the results of the classifiers generated using yeast benchmark data.

Discussion

Data analysis software typically quantifies peptides by aggregating the intensities of fragment ions extracted from MS/MS spectra in standard DIA mass spectrometry workflows. Previous methods for calculating precursor ion intensities by summing up fragment ions with TopN intensities may have overlooked the fact that multiple fragment ions interfere with each other in complex samples.

Here, we investigated a fragment ion level classifier and combination method based on the OpenSWATH workflow, hoping to simplify fragment ion sorting and automatically optimize peptide quantification using a subset of the precursor fragment ion population.

We implemented these algorithms in a Python package and wrapped each function of the workflow using Snakemake. We firstly used the SGD method to train the classifier from 12 features of fragment ions, and improved the performance of the classifier by ensuring a high precision score or dealing with noisy labels to provide a cleaner training set through different methods. This provided a more accurate discriminate score than the reference classifier for each fragment ion. We then extended the use of a group available Python package gscore algorithm for computing the q values of fragment ions for FDR control at the fragment ion level. After getting the candidate fragment ions, we tried different combinations of fragment ions to calculate precursor intensity. We combined these steps into a portable Python package that contains discriminant score calculation, q value calculation, FDR control on fragment ion levels, and different peptide quantification methods. It has user-friendly options and can be easily accessed to analyze specific datasets. Finally, to optimize combined fragment ions to provide quantitative information at the peptide and protein levels, we combined this Python package with the OpenSWATH and pyProphet workflows to form a simplified and optimized Snakemake pipeline.

On yeast benchmark proteomics data, we found an average accuracy improvement of 22.46% on the results of all machine learning models compared to the generalized OpenSWATH workflow. This suggests that optimizing the combination of fragment ions can improve the accuracy of protein quantification significantly.

Differences between classifiers

The average accuracy of the four classifiers on the yeast benchmark spike-in dataset is higher than the standard workflow. This demonstrates the power of our fragment ion ranking algorithm in combination with the target-decoy strategy.

Compared to other classifiers, The "initial" classifier achieves the highest accuracy within the ± 0.5 log2 fold change tolerance range on average, when using 0.01 as the cutoff of adjusted p values. This demonstrates the efficiency of the SGD training method on large datasets. Even without handling noisy labels, this training method achieves better protein quantification results than the standard workflows. However, other classifiers have higher ROC-AUC scores and higher accuracy on iRT peptide intensities than the "initial" classifier.

The "precision_90" classifier achieves good accuracy regardless of the measurement used. This shows that for the target-decoy classifier, it is necessary to have a minimum number of false positives within the results being predicted to be the target, that is, to ensure that the model has a high accuracy.

Based on the results, we would recommend "qvalue_cutoff" as the most practical method because its performance is stable regardless of the quantification method. This proves that q values calculation can be used to control false labels in the training space during the model training phase.

There are usually many noisy labels in PSMs results. As shown in Supplementary figure 4, the decision score distribution of the reference classifier has two peaks in the entire decision score interval on the spike-in dataset, and the first peak almost coincides with the decoy labels, which are considered to be the false targets, i.e. noisy labels. Low quality labels usually lead to low quality predictions, so when many false labels are treated as true labels, the accuracy of the model will be poor. Clean and accurate labels are important for accurate predictions. Computing q values to identify and remove noisy labels in the training dataset can generate a cleaner training set, which can effectively improve model precision. The q value cutoff used to filter false targets in the training set was 0.01, a smaller cutoff may result in higher accuracy on protein quantification result but may result in a loss of true targets in the training space, so a q value cutoff of 0.01 is recommended.

The "lnl" classifier significantly outperforms the baseline on the yeast benchmark spike-in dataset, but not as good as other classifiers. But on multi-species proteomics data, it achieved higher accuracy than the "initial" and "qvalue_cutoff" classifiers, demonstrating the potential of this approach on similarly complex datasets.

Differences between quantification methods

We also investigated the differences between different precursor calculation methods. The mean, median and sum strategies have similar results (different classifiers have different results) regardless of the measurement used, probably because in these methods, the peptide intensities are determined by using all fragment ions passing FDR control. When using the top3 strategy, the results of different classifiers are relatively consistent, which shows the stability of its performance. This suggests that further combining fragment ions passing FDR control appropriately can significantly affect the result of downstream protein quantification.

Among the four quantification methods, the ROC-AUC score of the top3 strategy is relatively lower on average than the other three methods. This shows that the results of the top3 strategy are more dependent on the cutoff value of adjusted p values than other methods. The top3 strategy achieved the highest accuracy over the entire log2 fold change tolerance range when using 0.01 as a cutoff of the adjusted p values, and it is more stable no matter what classifier it is combined with. We also investigated other topN methods and found that top3 performed better than top1 and top2 method, which means that there is a specific N that maximizes the accuracy of the topN method. We therefore recommend the top3 method as the default quantification method.

Additionally, when selecting N fragment ions in the topN method, we sort the fragment ions by q values rather than fragment ion intensities, so the final result depends on the classifier's decision scores rather than the fragment ion intensities. Previous methods using the N most abundant fragment ions for quantification may ignore the fact that the fragment ions passing FDR control may originate from multiple peptides. These shared fragment ions can be filtered by specific algorithms in future work. Sorting fragment ions by intensities in the topN method may significantly change the results after excluding shared fragment ions. Furthermore, in the topN method, when N is greater than 1, the FDR control at the fragment ion level will result in

uneven fragment ion numbers, which may introduce errors and needs to be addressed in the future.

Other methods such as mean and median methods cannot achieve the same accuracy as top3. We suspect this is because the mean and median intensities of fragment ions may not be able to represent peptide intensities. Fragment ions with smaller intensities will significantly reduce the peptide intensity. This will lead to an overestimation of fragment ions with low abundance and a severe underestimation of fragment ions with high abundance.

Possible improvements

Due to the large amount of data, this project has limited options in model training method. The total data points are 42531675, which makes the "hold-out cross-validation" and SGD methods the best and the only method. However, in general, k-fold cross-validation training method is more stable than hold-out at the cost of a lot of computation and training time on large datasets.

We found that the performance of the yeast benchmark proteomics classifier and the classifier generated using the four species low fold change proteomics data showed consistent performance on the four species low fold change proteomics data. This demonstrates the potential applicability of the classifier on other datasets, if one classifier works for all experiments, it will save the model training and improvement process. So, it is necessary to spend more time to improve the performance of the classifier in the future. This can be achieved by using new training algorithms, by optimizing features or adding new features. Building different layers and training a neural network model to change how different features are combined and compute their weights may achieve better results. In the selection of features, the contribution of some features to the model output varies across classifiers, while the contribution of other features to the model output remains consistent across classifiers (Supplementary figure 2). Shapley values can be used as a reference to filter features that have less effect on the model in all methods. The backpropagation process is another candidate, which allows the model to learn the weights of the input features. In addition, other manually calculated features, such as fragment ion interference scores, can also be added into the input features.

Conclusion

Previous standard DIA quantitative analysis workflows ignored the fact that fragment ions can interfere with each other in complex samples. In this project, we tested the possibility of using machine learning and algorithms for FDR control at the fragment ion level to pick the most optimal fragment ion population to improve the accuracy of peptide quantification. Using the algorithmic approaches we suggested in combination with machine learning to determine fragment ion ranks within a peak group, we can combine them in an optimal way to provide more precise peptide quantification and improve the accuracy of protein differential expression results significantly.

Acknowledgments

I would like to thank my supervisors for bringing me into this group. Thanks to Aaron for teaching me a lot. Thanks to Di for telling me the email address of the supervisor so I had the opportunity to join this group; thanks to Hong for showing the operation of the mass spectrometer to me; thanks to CK, Marc and Carlos for answering questions

about mass spectrometry; thanks to Tommy for inviting me to lunch every day; thanks Simon, you are a good friend; thanks to Fillip for teaching me how to make coffee; thanks to Erik for the daily wordle. I wish you all the best.

To myself: Even after graduation, you are still a student of life, remember to always learn. There are good time and bad time in life. Remember that it is always fluctuating, and when things are going well, don't forget to have a sense of crisis. In difficult times, please hold on to hope. And, instead of learning to wait, now is always the perfect time to do whatever you want.

Supplementary

materials.pdf

References

1. Zhu Y. MASS SPECTROMETRY BASED PROTEOMICS: DATA ANALYSIS AND APPLICATIONS;.

2. Doerr A. Mass spectrometry-based targeted proteomics;10(1):23–23. doi:10.1038/nmeth.2286.

3. Ludwig C, Gillet L, Rosenberger G, Amon S, Collins BC, Aebersold R. Data-independent acquisition-based SWATH-MS for quantitative proteomics: a tutorial;14(8):e8126. doi:https://doi.org/10.15252/msb.20178126.

4. Tsou CC, Avtonomov D, Larsen B, Tucholska M, Choi H, Gingras AC, et al. DIA-Umpire: comprehensive computational framework for data-independent acquisition proteomics;12(3):258–264. doi:10.1038/nmeth.3255.

5. Picotti P, Aebersold R. Selected reaction monitoring-based proteomics: workflows, potential, pitfalls and future directions;9(6):555–566. doi:10.1038/nmeth.2015.

6. Li KW, Gonzalez-Lozano MA, Koopmans F, Smit AB. Recent Developments in Data Independent Acquisition (DIA) Mass Spectrometry: Application of Quantitative Analysis of the Brain Proteome;13. doi:10.3389/fnmol.2020.564446.

7. Noor Z, Adhikari S, Ranganathan S, Mohamedali A. Quantification of Proteins From Proteomic Analysis. In: Ranganathan S, Gribskov M, Nakai K, Schönbach C, editors. Encyclopedia of Bioinformatics and Computational Biology. Oxford: Academic Press; 2019. p. 871–890. Available from: `https://www.sciencedirect.com/science/article/pii/B9780128096338206778`.

8. Lindemann C, Thomanek N, Hundt F, Lerari T, Meyer HE, Wolters D, et al. Strategies in relative and absolute quantitative mass spectrometry based

proteomics. Biological Chemistry. 2017;398(5-6):687–699. doi:doi:10.1515/hsz-2017-0104.

9. Neilson KA, Ali NA, Muralidharan S, Mirzaei M, Mariani M, Assadourian G, et al. Less label, more free: Approaches in label-free quantitative mass spectrometry. PROTEOMICS. 2011;11(4):535–553. doi:https://doi.org/10.1002/pmic.201000553.

10. Välikangas T, Suomi T, Elo LL. A systematic evaluation of normalization methods in quantitative label-free proteomics. Briefings in bioinformatics. 2018;19:1–11.

11. Willforss J, Chawade A, Levander F. NormalyzerDE: Online Tool for Improved Normalization of Omics Expression Data and High-Sensitivity Differential Expression Analysis. J Proteome Res. 2019;18(2):732–740. doi:10.1021/acs.jproteome.8b00523.

12. Huber W, von Heydebreck A, Sültmann H, Poustka A, Vingron M. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. Bioinformatics (Oxford, England). 2002;18 Suppl 1:S96–104.

13. Escher C, Reiter L, MacLean B, Ossola R, Herzog F, Chilton J, et al. Using iRT, a normalized retention time for more targeted measurement of peptides. Proteomics. 2012;12:1111–21.

14. Searle BC, Pino LK, Egertson JD, Ting YS, Lawrence RT, MacLean BX, et al. Chromatogram libraries improve peptide detection and quantification by data independent acquisition mass spectrometry. Nature Communications. 2018;9(1):5128. doi:10.1038/s41467-018-07454-w.

15. Reiter L, Rinner O, Picotti P, Hüttenhain R, Beck M, Brusniak MY, et al. mProphet: automated data processing and statistical validation for large-scale SRM experiments. Nature Methods. 2011;8(5):430–435. doi:10.1038/nmeth.1584.

16. Demichev V, Messner CB, Vernardis SI, Lilley KS, Ralser M. DIA-NN: neural networks and interference correction enable deep proteome coverage in high throughput. Nature methods. 2020;17:41–44.

17. Elias JE, Gygi SP. Target-decoy search strategy for mass spectrometry-based proteomics. Methods in molecular biology (Clifton, NJ). 2010;604:55–71.

18. Benjamini Y, Hochberg Y. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. Journal of the Royal Statistical Society: Series B (Methodological). 1995;57(1):289–300. doi:https://doi.org/10.1111/j.2517-6161.1995.tb02031.x.

19. Käll L, Storey JD, MacCoss MJ, Noble WS. Posterior Error Probabilities and False Discovery Rates: Two Sides of the Same Coin. J Proteome Res. 2008;7(1):40–44. doi:10.1021/pr700739d.

20. Käll L, Canterbury JD, Weston J, Noble WS, MacCoss MJ. Semi-supervised learning for peptide identification from shotgun proteomics datasets. Nature Methods. 2007;4(11):923–925. doi:10.1038/nmeth1113.

21. The M, MacCoss MJ, Noble WS, Käll L. Fast and Accurate Protein False Discovery Rates on Large-Scale Proteomics Data Sets with Percolator 3.0. Journal of the American Society for Mass Spectrometry. 2016;27(11):1719–1727. doi:10.1007/s13361-016-1460-7.

22. Röst HL, Rosenberger G, Navarro P, Gillet L, Miladinović SM, Schubert OT, et al. OpenSWATH enables automated, targeted analysis of data-independent acquisition MS data. Nature Biotechnology. 2014;32(3):219–223. doi:10.1038/nbt.2841.

23. Teleman J, Röst HL, Rosenberger G, Schmitt U, Malmström L, Malmström J, et al. DIANA—algorithmic improvements for analysis of data-independent acquisition MS data. Bioinformatics. 2014;31(4):555–562. doi:10.1093/bioinformatics/btu686.

24. Pham TV, Henneman AA, Jimenez CR. iq: an R package to estimate relative protein abundances from ion quantification in DIA-MS-based proteomics. Bioinformatics. 2020;36(8):2611–2613. doi:10.1093/bioinformatics/btz961.

25. Bilbao A, Zhang Y, Varesio E, Luban J, Strambio-De-Castillia C, Lisacek F, et al. Ranking Fragment Ions Based on Outlier Detection for Improved Label-Free Quantification in Data-Independent Acquisition LC–MS/MS;14(11):4581–4593. doi:10.1021/acs.jproteome.5b00394.

26. Teo G, Kim S, Tsou CC, Collins B, Gingras AC, Nesvizhskii AI, et al. mapDIA: Preprocessing and statistical analysis of quantitative proteomics data from data independent acquisition mass spectrometry. Journal of Proteomics. 2015;129:108–120. doi:10.1016/j.jprot.2015.09.013.

27. Mitchell T. Machine Learning. McGraw Hill; 1997.

28. Improving First and Second-Order Methods by Modeling Uncertainty; 2012.

29. Kumari R, Srivastava SK. Machine learning: A review on binary classification. International Journal of Computer Applications. 2017;160(7).

30. Ruder S. An overview of gradient descent optimization algorithms; 2016. Available from: `https://arxiv.org/abs/1609.04747`.

31. Kohl M. Performance Measures in Binary Classification. International Journal of Statistics in Medical Research. 2012;1:79–81. doi:10.6000/1929-6029.2012.01.01.08.

32. Bruderer R, Bernhardt OM, Gandhi T, Xuan Y, Sondermann J, Schmidt M, et al. Optimization of Experimental Parameters in Data-Independent Mass Spectrometry Significantly Increases Depth and Reproducibility of Results. Molecular cellular proteomics : MCP. 2017;16:2296–2309.

33. Martens L, Chambers M, Sturm M, Kessner D, Levander F, Shofstahl J, et al. mzML–a community standard for mass spectrometry data. Molecular cellular proteomics : MCP. 2011;10(20716697):R110.000133–R110.000133. doi:10.1074/mcp.R110.000133.

34. Hulstaert N, Shofstahl J, Sachsenberg T, Walzer M, Barsnes H, Martens L, et al. ThermoRawFileParser: Modular, Scalable, and Cross-Platform RAW File Conversion;19(31755270):537–542. doi:10.1021/acs.jproteome.9b00328.

35. Adusumilli R, Mallick P. Data Conversion with ProteoWizard msConvert. Methods in molecular biology (Clifton, NJ). 2017;1550:339–368.

36. Kurtzer GM, Sochat V, Bauer MW. Singularity: Scientific containers for mobility of compute. PLOS ONE. 2017;12(5):1–20. doi:10.1371/journal.pone.0177459.

37. Rosenberger G, Bludau I, Schmitt U, Heusel M, Hunter CL, Liu Y, et al. Statistical control of peptide and protein error rates in large-scale targeted data-independent acquisition analyses. Nature Methods. 2017;14(9):921–927. doi:10.1038/nmeth.4398.

38. Köster J, Rahmann S. Snakemake–a scalable bioinformatics workflow engine;28(19):2520–2522. doi:10.1093/bioinformatics/bts480.

39. Röst HL, Liu Y, D'Agostino G, Zanella M, Navarro P, Rosenberger G, et al. TRIC: an automated alignment strategy for reproducible protein quantification in targeted proteomics. Nature Methods. 2016;13(9):777–783. doi:10.1038/nmeth.3954.

40. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 2011;12:2825–2830.

41. Käll L, Storey JD, Noble WS. QVALITY: non-parametric estimation of q-values and posterior error probabilities. Bioinformatics (Oxford, England). 2009;25:964–6.

42. Northcutt CG, Jiang L, Chuang IL. Confident Learning: Estimating Uncertainty in Dataset Labels. 2019;doi:10.48550/ARXIV.1911.00068.

43. Northcutt CG, Jiang L, Chuang IL. Confident Learning: Estimating Uncertainty in Dataset Labels. 2019;doi:10.48550/ARXIV.1911.00068.

44. Sundararajan M, Najmi A. The many Shapley values for model explanation; 2019. Available from: `https://arxiv.org/abs/1908.08474`.